

String operations (string s)

<code>s.count(substring)</code>	Count occurrences
<code>s.find(substring)</code>	Index of first occurrence
<code>s.join(sequence)</code>	Concatenate sequence
<code>s.split([delimiter])</code>	Split into list

List operations (list l, element e)

<code>l.append(e)</code>	Add e
<code>l.remove(e)</code>	Remove e
<code>l.pop(e)</code>	Remove and return e
<code>l.count(e)</code>	Count occurrences
<code>l.reverse()</code>	Reverse l
<code>l.sort()</code>	Sort l

File operations (file f)

<code>f = open(path)</code>	Open f
<code>f.read()</code>	Read f
<code>f.readline()</code>	Read line from f
<code>f.readlines()</code>	Return list of lines in f
<code>f.write(s)</code>	Write s to f
<code>f.close()</code>	Close f

Object-oriented

<code>class Person:</code>	Class definition
<code>x = Person(age, height)</code>	Object creation
<code>x.age</code>	Field access
<code>x.birthday()</code>	Method access

Operators

Assignment	=
Arithmetic	+, -, *, /, %
Comparison	>, >=, <, <=, ==, !=
Logical	not, and, or

Dictionary operations (dict d, key k)

<code>d.clear()</code>	Clear d
<code>d.get(k)</code>	Return d[k]
<code>d.keys()</code>	Return keys in d
<code>d.values()</code>	Return values in d
<code>d.items()</code>	Return key-value pairs in d

Control Flow

<code>if(cond): <code> else: <code></code>	If-else
<code>if(cond1): <code> elif(cond1): <code> else: <code></code>	If-elif-else
<code>for i in range([start], stop, [step]): <code></code>	For loop over range
<code>for i in items: <code></code>	For loop over iterable
<code>while(condition): <code></code>	While loop
<code>break</code>	Exit first enclosing loop
<code>continue</code>	Skip to next iteration

Module import

```
import module

from module import submodule
```

Useful standard library modules

<code>math, numpy, scipy</code>	Math
<code>matplotlib</code>	Graph plotting
<code>random</code>	Random generators
<code>datetime</code>	Date and time
<code>timeit</code>	Performance
<code>re</code>	Regular expressions
<code>os</code>	OS interaction
<code>sys</code>	stdin, stdout, stderr
<code>urllib</code>	Internet access
<code>zlib</code>	Data compression

In-built functions

<code>int(), float(), str(), bool()...</code>	Type casting
<code>len()</code>	Length
<code>min(), max()</code>	Minimum / Maximum
<code>pow(x, y, [z])</code>	X to the power Y [mod Z]
<code>range(start, stop, [step])</code>	Ordered list
<code>input(), print()</code>	Console Input/Output
<code>filter(function, array)</code>	Filter array
<code>map(function, array)</code>	Map function onto array
<code>id(object)</code>	Unique object ID
<code>round(n, [x])</code>	Round n [x decimal places]

